REMARKS

The Office Action dated August 29, 2005, has been carefully reviewed and the following remarks are submitted in response thereto. Claims 1-12 and 21-36 are pending in the application.

The rejection of claims 1-12 and 21-36 under 35 USC §112, first paragraph, as failing to comply with the written description requirement is respectfully traversed.

The rejection argues that the specification does not disclose "a plurality of separately secured remote applications" or similar references to first and second separately secured applications. In actuality, the plurality of separately secured applications are expressly described throughout the specification. The Description of the Prior Art on page 1 discusses the problem that "computer users who wish to gain access to more than one ... application ... during a computer session to repeatedly re-enter their user IDs ... each time they attempt to transfer from ... one application or program to another." On page 3, it is disclosed that the "application servers 14 are coupled with the user computers 12 via the communications network 22 and are provided for running applications on behalf of the user computers." Thus, the applications run by a computer user are remote applications. On pages 5 and 6, it is described how a "user first launches some application or program in a conventional manner" and how the "user next logs into the selected authorization server 16 using account or ID information". Thus, the remote application is a secured application. Page 7 describes that "when the user attempts to access other applications ... while he or she is still logged into the system, these other applications may reference the Session ID ... for authorization purposes related to the new applications." Thus, the specification teaches a plurality of separately secured remote applications.

The rejection also argues that the specification does not indicate how to store a link or retrieve a link. The claims recite storing security information in an object within a dynamic directory on a directory server and then storing a link to the object on the user computer. On page 6, the specification teaches:

> The authorization server 16 then copies or links the

> Session ID or some derivative thereof to something on the user's computer 12 such as a cookie, shared application memory, or the computer's network address. It is important only that other applications launched by the user from the user computer be able to read or otherwise determine this Session ID by accessing something on the user's computer.

One skilled in the art would reasonably understand that in the act of linking the Session ID to something on the user's computer, a link which may have the form of a cookie is created and stored. Once created, other applications may read (retrieve) it. Thus, the claims are fully supported and enabled, and the rejection under 35 USC 112 should be withdrawn.

The rejection of claims 1-4, 7-10, 21, 24, 27, 29, 30, 32, 34, and 35 under 35 USC 102(e) as being anticipated by Alegre et al is respectfully traversed. In the method and system of claims 1, 7, 27, and 32, an object associated with the Session ID is stored dynamically in a directory on a directory server coupled with the authorization server. The directory server permits other computer applications launched by the computer user to reference the Session ID on the user's computer. The user is authenticated and authorized to the first secured computer application to be launched by interacting with an authorization server. The user is authenticated and authorized to a second separately-secured computer application by accessing the object for the computer user on the directory server rather than requiring further interaction with the authorization server. The ability of additional applications to authenticate or authorize directly with the directory server achieves important advantages such as reducing network overhead.

Applicant respectfully points out that Alegre et al fails to teach all the claimed limitations, either expressly or implicitly. Alegre et al neither shows nor suggests separately-secured computer applications that are remotely launched by a user. Rather than authenticating and authorizing a user with respect to separately secured applications, Alegre et al creates a session key that is stored at a client browser and is used to access a trusted network. As opposed to authenticating a user to a particular application, Alegre et al requires every message transmitted from the

user to the network to be authenticated. Whenever the user accesses the trusted network during the session, the session key must be transmitted with the access request (col. 3, line 67, to col. 4, line 7). Thus, user authentication is checked for each and every individual remote access request by the user. The session key must be transmitted and checked with every incoming access request from the user, resulting in very high network overhead which is avoided by the present invention. Alegre et al does not have any teaching of authenticating a user to a remote application on an application server, as is required by the present claims.

Alegre et al has no teaching whatsoever of multiple applications that each requires its own separate authorization. Therefore, there is likewise no teaching of using a directory to store an object accessed by more than one application for purposes of authentication.

The rejection relies on Alegre et al at column 8, lines 16-27, to allegedly show multiple applications. The relevant portion states:

> In addition to the check against the access profile received from key server 234, applications requiring extra fine grained access control may use the UID received from key server 234 in combination with a local data base of access rules (not shown) to implement additional access policies.

Besides not providing any description of access rules, Alegre et al does not provide any teaching to show what the "additional access policies" are. There is no teaching or suggestion that the user is authenticated or authorized based on information stored in the key server. Further input (e.g., entry of a password - which is avoided by the present invention) might be expected. Alegre et al only invites speculation regarding this point. On the other hand, it is clear that Alegre et al has no teaching of a second application accessing an object created when an earlier application was launched. Therefore, claims 1-4, 7-10, 21, 24, 27, 29, 30, 32, 34, and 35 are allowable over Alegre et al.

The rejection of claims 5, 6, 11, 12, 31, and 36 under 35 USC 103(a) as being unpatentable over Alegre et al in view of Hartman et al is respectfully traversed. Hartman fails to correct for the deficiencies in Alegre. Therefore, claims 5, 6, 11, 12,

31, and 36 are allowable.

The rejection of claims 22, 23, 25, 26, 28, and 33 under 35 USC 103(a) as being unpatentable over Alegre et al in view of Blanco et al is respectfully traversed. Blanco et al does not use LDAP or X.500 to access objects having the limitations recited in the present claims. Thus, Blanco et al fails to correct for the deficiencies in Alegre et al, and claims 22, 23, 25, and 26 are allowable.

In view of the foregoing remarks, claims 1-12 and 21-36 are respectfully submitted to be in condition for allowance. Favorable action is respectfully solicited.

Respectfully submitted,

Mark L. Mollon
Attorney for Applicant(s)
Reg. No. 31,123

Dated: November 4, 2005
MacMillan, Sobanski & Todd, LLC
One Maritime Plaza, Fourth Floor
720 Water Street
Toledo, Ohio 43604
(734) 542-0900
(734) 542-9569 (fax)